Ikezoye, Vance

From: Franck Chastagnol [fchastagnol@youtube.com]
Sent: Friday, September 22, 2006 10:48 AM
To: Schrempp, Jim
Cc: Ikezoye, Vance; David Estrada
Subject: Re: YouTube/AudibleMagic integration

Hi Jim,

> With the current deal, the number of servers becomes our issue, so we should remove that from the SLA.

makes sense

> As to concurrent processes, when we go live you can do 16 concurrent, as your forecast of rates goes up we can increase that number for you. Can you send me the first forecast of transaction rates?

16 concurrent txns makes sense.
i can work on forecast for daily volume of match API and add API calls. would a 3 months after launch and 6 months after launch forecasts suffice ?

In terms of SLA, here is what I suggest we put in the contract, let me know if you are ok with this
- overall availability of AM system: <need number from AM>
- match API SLAs (see requirements below, this is what we had discussed)
- add API SLAs (see requirements below. we had not discussed it but I hope it is reasonable for you)

I think it may make sense to add the requirements we discussed together (see below) as an exhibit to the contract ?

Thanks,
Franck

Requirements for integration with Audible Magic (AM)¶

Phase 1 (launch)¶

Database setup¶

- AM should host and setup 2 reference fingerprint databases:
  - one populated with Warner content (referenced as 'WarnerDB' hereafter)
  - one populated by YouTube via the new fingerprint add API (this DB is referenced as 'YouTubeDB' hereafter)

DATE: 12/10/08 EXHIBIT#
DEPONENT: Chastagnol 13
CASE: Viacom, et al., v. YouTube, et al., The Football Association Premier League, et al., v. YouTube, et al., Case Nos. 07-CV-2203 and 07-CV-3582
A. Ignacio Howard, CLR, RPR, CSR No. 9830

- YouTubeDB requirements:
  - should be able to store 200,000 (two hundred thousands) fingerprints at launch
  - should have the capability to scale upon YouTube explicit request up to several Millions

- Fingerprinting match mode
  - WarnerDB should be setup in "File identification" mode
  - YouTubeDB should be setup in "File identification" mode

Fingerprint Match API¶

- A single match API call should have ability to query against WarnerDB and/or YouTubeDB
  - There will be a paramater "target_db" in the URL of the HTTP request that indicates the target database: WarnerDB, YouTubeDB, or both WarnerDB and YouTubeDB
  - The possible values for target_db are:
    - "db1" = WarnerDB
    - "db2" = YouTubeDB
    - "all" = both WarnerDB and YouTubeDB
- If target_db parameter is not present in URL, default value "all" should be assumed: request should go against both WarnerDB and YouTubeDB
- If target_db is "all", fingerprint lookup algorithm should:
  - lookup first on WarnerDB, then on YouTubeDB
  - return as soon as a match is found. e.g. if a match is found on WarnerDB, there is not need to search on YouTubeDB

- match API should return the following standard meta-data information in its XML response (content owner is Warner for Phase I):
  - ISRC, if supplied by content owner
  - GRiD, if supplied by content owner (Warner specific)
  - Product ID, if suplied by content owner (Warner specific)
  - Artist name
  - Track title
  - Album name
  - Releasing Label
  - Genre , if supplied by content owner

- match API should return additional meta-data (this may require custom work on AM side):
  - <supplier> field, indicating the source of content (Warner, Sony, Universal, etc...)
  - <policy>: block content or share revenues. policy should be set by Warner and sent to AM to store as a business rule in WarnerDB. default policy will be "block". ' **This is contingent upon Warner passing this meta-data to AM**
  - <content_type> field, indicating type of content: Music or Music Video. **This is contingent upon Warner passing this meta-data to AM**

- AMItemID
  - The AMItemID returned in case of a match against YouTubeDB should:
    - have its 3 last characters be as "YTB" (or any specific 3 chars defined by AM)
    - have the other characters (max 40 chars) representing the custom id passed as part of the add fingerprint API (see below)

- SLAs
  - match API performance:

        - the response time is measured at ingress point of the AM network
        - average response time should not exceed 8 secs
        - 90th percentile response time within 12 secs
        - AM should be able to handle concurrent transactions. At launch, YouTube will have up to 16 concurrent transactions. As volume grows, it will be possible to increase this number.
    - match API volume
        - At launch, AM should be able to handle 200,000 (two hundred thousands) match API calls / day.
        - Based on forecast volume numbers provided by YouTube, AM should be able to scale

Fingerprint Add API¶

- AM should expose a new API for YouTube to be able to incrementally add a new reference fingerprint to YouTubeDB
- Each day, new reference fingerprints received by 11:00am should be uploaded to YouTubeDB by 18:00am

- interface definition:
    - HTTP POST
    - XML body
    - XML will include:
        - binary reference fingerprint, generated using AM MFCBR C library
        - custom_id: max 32 alphanumeric chars, returned in case of a match by the match API

- SLAs (**Still to be discussed with AM**)
    - add API performance:
        - the response time is measured at ingress point of the AM network
        - average response time should not exceed 8 secs
        - 90th percentile response time within 12 secs
        - AM should be able to handle concurrent transactions.
    - add API volume
        - At launch, AM should be able to handle 5,000 (five thousands) add API calls / day
        - Based on forecast volume numbers provided by YouTube, AM should be able to scale

Reports¶

- AM should provide a high level weekly report.
    - Report for previous week should be sent to YT by email every Monday.
    - Report should contain general information including (but not limited to):
        - total number of API match queries made by YouTube
        - general stats on performance and system load (average response time, 90th percentile response time, average and peak server load, ...). Stats at AM network.
        - number of fingerprints in WarnerDB and YouTubeDB
- Upon YouTube request, AM should be able to provide a detailed report of the data present in YouTubeDB including (but not limited to) list of custom_id

Phase 2 (date TBD)¶

Fingerprint Delete API¶

- API allowing to delete a particular reference fingerprint from YouTubeDB
- HTTP POST, XML request/response
- input: custom_id
- output: success/failure

Fingerprint Lookup API¶

- API allowing to query for the existence of a fingerprint in the WarnerDB or YouTubeDB
- HTTP POST, XML request/response
- input: custom_id or ISRC
- output: present yes/no, some meta-data (fingerprint length, time uploaded, etc...)
